Specifying BYOD Policies With Authorisation Logic

Joseph Hallett and David Aspinall

University of Edinburgh

Abstract

BYOD policies are imprecisely specified using natural language. This creates ambiguity, may hide specification problems, and makes comparison harder. We show how we can use AppPAL (an authorisation logic) to formalise three BYOD policies. We identify potential problems in all three policies. This suggests authorisation logic may help in policy specification and enforcement.

1 Introduction

Employees bring their personal devices to work. Corporate IT departments can find this challenging: they want to secure their networks but have limited control over a personal device's software and configuration. To ensure compliance IT departments publish mobile device policies. Employees must agree to follow them if they want to use their personal devices at work. Policies specified using natural language can contain ambiguities. This makes compliance checking hard as automatic methods are hard to use with these policies.

Logics of authorisation are languages used to express permissible actions. Their uses include access control [6], and identifying problems in electronic health records [8] amongst other applications. *AppPAL* [3] instantiates Becker et al.'s SecPAL [1] language to express mobile app installation policies. It makes use of delegation relationships and uses external constraints to formalise policies. AppPAL policies define predicates which can affect application installation (*isInstallable*) or operation; they may be enforced by application rewriting or a modified version of the mobile platform security manager. We believe AppPAL has applications to mobile device BYOD policies as well.

AppPAL is designed to be readable; we introduce the language alongside example excerpts from three BYOD including: a security policy from the SANS institute to aid companies in developing their own rules [2], an NHS hospital trust [4], and a policy used at The University of Edinburgh [5]. We show how AppPAL can specify BYOD policies making them precise, identifying problems and aiding comparison.

2 Translating Corporate BYOD Policies

Many BYOD policies describe the apps usable in the workplace. The Edinburgh policy recommends apps come from reputable sources (though the meaning of *reputable* is left undefined). The SANS policy encourages the IT department to maintain black-lists and white-lists. In fact the SANS policy recommends this multiple times in their document.

Edinburgh: *"Only download applications (apps) or other software from reputable sources."*

```
'edinburgh−policy' says Store can-say App isInstallable if Store sells(App), Store isReputable.
```

SANS: *"The IT Department maintains a list of allowed and unauthorised applications and makes them available to users on the intranet."*

```
'sans−policy' says 'it−department' can-say App isInstallable.
```

SANS: *"Only approved third party applications can be installed on handhelds. The approved list can be obtained by contacting the IT department, or should be available on the intranet."*

```
'sans−policy' says 'it−department' can-say App isInstallable.
```

The NHS policy distinguishes between *clinical* and *non-clinical* apps; with both requiring approval through a different group before a committee gives final approval. In practice these rules are irrelevant: no group has ever approved an app. This is, perhaps, unsurprising as 82% of Android medical apps have basic privacy concerns [7, 9].

NHS: *"Apps for work usage must not be downloaded onto corporately issued mobile devices (even if approved on the NHS apps store) unless they have been approved through the following Trust channels: Clinical apps; at the time of writing there are no apps clinically approved by the Trust for use with patients/clients. However, if a member of staff believes that there are clinical apps* [· · ·] *ratification should be sought via the Care and Clinical Policies Group.* [· · ·] *Business apps; at the time of writing there are no business (i.e., non-clinical) apps approved by the Trust for use other than those preloaded onto the device at the point of issue. However, if a member of staff believes that there are apps* [· · ·] *ratification of the app must be sought via the Management of Information Group (MIG).* [· · ·] *Following approval through Care and Clinical Policies and/or MIG, final approval will be required through Integrated Governance Committee."*

```
'nhs−policy' says App isInstallable if App isUsable, App isFinallyApproved.
'nhs−policy' says 'cacpg' can-say App isUsable if App isUsableClinically.
'nhs−policy' says 'mig' can-say App isUsable if App isUsableNonClinically.
'nhs−policy' says 'igc' can-say App isFinallyApproved.
```

All three policies use delegation to decide is an app is installable. By translating the policies into AppPAL problems become apparent: the Edinburgh policy is *incomplete* as it lacks a means to determine what stores were reputable, the SANS policy *contains duplicated rules*, and the NHS policy is incomplete and contains *redundant statements* as no app has ever been approved for use. For all three policies the AppPAL rule is no longer than the natural language policy, and in the case of the NHS policy considerably shorter and clearer.

Similarities appear when looking at how devices connect to networks. The Edinburgh policy recommends disabling automatic access to open unsecured Wi-Fi networks (though again it does not specify what *secure* actually means). In contrast, the SANS policy says what security features are required and that, whilst secure access points are preferred, an open access point is acceptable if used with a VPN.

Edinburgh: *"Control your devices connections by disabling automatic connection to open, unsecured Wi-Fi networks."*

```
'edinburgh−policy' says Device canConnectTo(AP) if AP isSecure.
```

SANS: *"If mobile workers do require connectivity through public, open, or untrusted WLAN, then users MUST use WLANs using, if available and in this order: WPA(2) encryption, WEP 256 bits (or 128 bits), or finally open networks if nothing else is available. Users connected to data networks in an open environment MUST use a VPN connection."*

```
'sans−policy' says Device canConnectTo(AP) if AP isSecure.
'sans−policy' says AP isSecure if AP canUseWPA2.
'sans−policy' says AP isSecure if AP canUseWEP256.
'sans−policy' says AP isSecure if AP canUseWEP128.
'sans−policy' says Device canConnectTo(AP) if Device hasVPN(VPN) where connect(VPN) = true.
```

By analysing the similarities in policies a framework of BYOD predicates could be defined that would allow policies to be compared using standard terms *precisely*.

AppPAL can describe complex scenarios in policies. The NHS policy describes when a clinician can use their phone's camera as part of patient care. The rules require obtaining consent (delegation to the patient) as well as clinical approval (delegation to the clinician).

As well as helping clarify the policy, the AppPAL rules can aid with compliance—a proof an AppPAL policy was satisfied might act as evidence that proper procedure was followed.

NHS: *"Some mobile devices have the ability to take photographs/videos. This function should not be used for photographs/videos of an individuals care and treatment unless the device has encryption enabled and it is clinically appropriate to do so. If the photography/video facility is used as part of the recording of an individuals care and treatment, the device user must ensure that the consent of the individual has been collected prior to taking any photograph/video.* [$\cdots$] *A record of the consent must be entered into the individuals care record."*

```
'nhs−policy' says Device canPhotograph(Patient) if Device isEncrypted, Patient canBePhotographed.
'nhs−policy' says ' clinician ' can-say Patient canBePhotographed
  if Patient hasClinicalNeedForPhotograph, Patient hasConsentedToPhotograph
  where recordedICRConsentForPhotograph(Patient) = true.
'nhs−policy' says ' clinician ' can-say Patient hasClinicalNeedForPhotograph.
'nhs−policy' says Patient can-say Patient hasConsentedToPhotograph if Patient canConsent.
```

3 Current Status

So far we have developed AppPAL policies describing each of the BYOD policies. We have developed tools to identify when AppPAL policies may be incomplete, and aim to extend them to inconsistency and redundancy checks in future work. For example, when checking the *isInstallable* predicate in the NHS policy our completeness checker spots the missing statements from the delegated to groups, and that the *isInstallable* rule is unsatisfiable.

```
The following assertions are unsatisfiable:
  'nhs−policy' says App.1 isInstallable if App.1 isUsable, App.1 isFinallyApproved.
These decisions may be derivable but we lack statements from the delegated party:
  (via 'mig') 'nhs−policy' says * isUsable
  (via 'igc') 'nhs−policy' says * isFinallyApproved
  (via 'cacpg') 'nhs−policy' says * isUsable
```

As the BYOD policies are informal we should question the accuracy of our formalisation. Future work will look at developing a testing process as part of the requirements capture, and working with policy owners to check that rules AppPAL postulates as true indeed match the author's intentions. In general, BYOD policies are at an early stage. As policies grow and become more intricate, we believe that tools like AppPAL will help ensure they are well specified.

References

[1] M.Y. Becker, C Fournet, and A D Gordon. SecPAL: Design and semantics of a decentralized authorization language. *Computer Security Foundations*, 2006.

[2] N.R.C. Guérin. Security Policy for the use of handheld devices in corporate environments, *SANS*.

[3] J. Hallett and D. Aspinall. AppPAL for Android. *Engineeering Secure Software and Systems*, 2016.

[4] G. Kennington, et al. Mobile Devices Policy, *Torbay and Southern Devon Health Care NHS Trust*.

[5] D. Williamson, A. Grzybowski, and S. Graham. BYOD Policy: Use of Personally Owned Devices for University Work, *University of Edinburgh*.

[6] M. Abadi. Logic in access control, *Logic in Computer Science*, 2003.

[7] S.R. Blenner, et al. Privacy Policies of Android Diabetes Apps and Sharing of Health Information *Journal of the American Medical Association*, 2016.

[8] M.Y. Becker, and P. Sewell, Cassandra: flexible trust management, applied to electronic health records, *Computer Security Foundations*, 2004.

[9] K. Knorr, and D. Aspinall. Security Testing for Android mHealth Apps *Software Testing, Verification and Validation Workshops*, 2015.